



The **LISA International Science Community** (LISC) is an informal researcher network maintained by the LISA International Science Team for the purpose of exchanging information about LISA with the wider science community.

**[www.lisascience.org](http://www.lisascience.org)**

You are encouraged to join\* the LISC at its **web portal**, [www.lisascience.org](http://www.lisascience.org), which features **LISA news**, introductory and advocacy **resources**, **links**, and **discussions**.

\*If you have already completed a preliminary registration (name and e-mail) on [www.lisascience.org](http://www.lisascience.org) upon receiving the first issue of the LISA newsletter, you should soon receive an e-mail with instructions on completing your registration.

6th LISA Symposium  
June 23, 2006

# A Mock LISA Data Challenge How-To

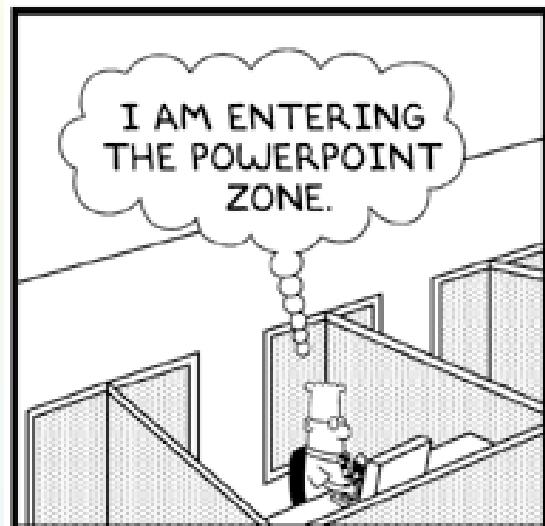
For the **Mock LISA Data Challenge Taskforce**:  
K. Arnaud, S. Babak, J. Baker, M. Benacquista,  
J. Centrella, N. Cornish, C. Cutler, S. Larson,  
B. Sathyaprakash, M.V., A. Vecchio, J.-Y. Vinet

Michele Vallisneri

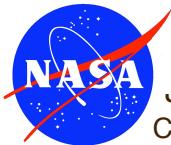




Jet Propulsion Laboratory  
California Institute of Technology

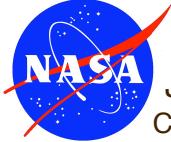


© Scott Adams, Inc./Dist. by UFS, Inc.



## The MLDC taskforce produced

- Challenge definitions
- Standard source models for galactic binaries, BBHs, EMRIs
- New cross-tested versions of simulators
- A *pseudo-LISA* specification
- Noise and TDI conventions
- An XML file format with read/write libraries
- Dataset generation pipelines
- Collaboration and documentation tools (website, wiki, SourceForge)
- A challenge evaluation roadmap



## Galactic binaries

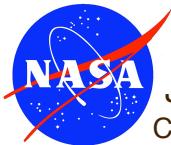
- Circular, nonevolving (monochromatic)
- Measured positions and period for known verification binaries; other parameters drawn randomly
- Otherwise, drawn from Gijs' population-dynamics simulation

## SMBH binaries

- Circular, adiabatic, 2PN inspiral, tapered after ISCO
- Parameters drawn from uniform random distributions, with constraint on S/N

## EMRIs

- Cutler-Barack (analytic kludge)
- Parameters drawn from uniform random distributions, with constraint on S/N



Jet Propulsion Laboratory  
California Institute of Technology

# THE LISA SIMULATOR

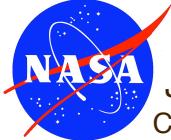
Neil Cornish and Louis Rubbo, PRD 67, 022001 (2003)  
<http://www.physics.montana.edu/lisa>

- Λ Computes LISA phase delays by integrating GW strain along photon paths
- Λ Written in C; reads/writes lisaXML
- Λ File-oriented workflow; challenge edition = 2.1.1

## Synthetic LISA (synthLISA for friends)

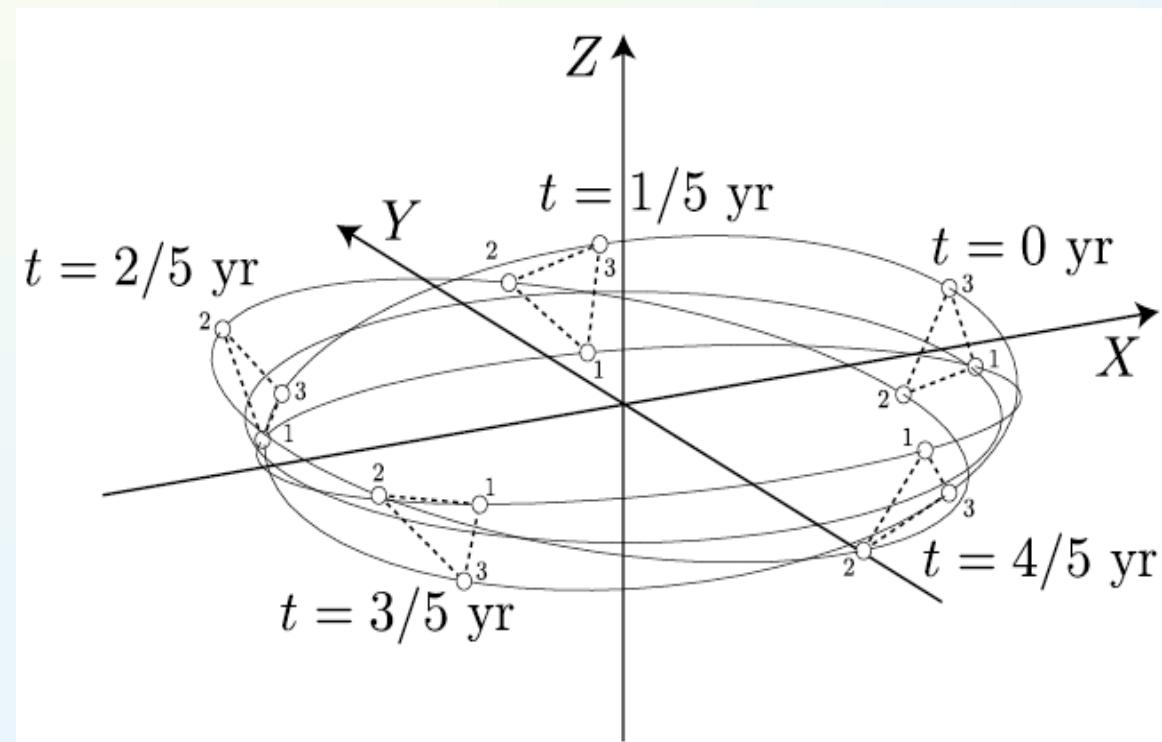
MV, PRD 71, 022001 (2005)  
<http://www.vallis.org/syntheticlisa>

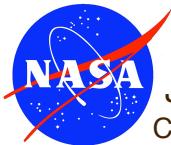
- Λ Computes LISA frequency fluctuations (time-delayed differences of strains)
- Λ C++ library, Python frontend and high-level functions; reads/writes lisaXML
- Λ Object-oriented workflow; challenge edition = 1.3.1



## pseudo-LISA geometry

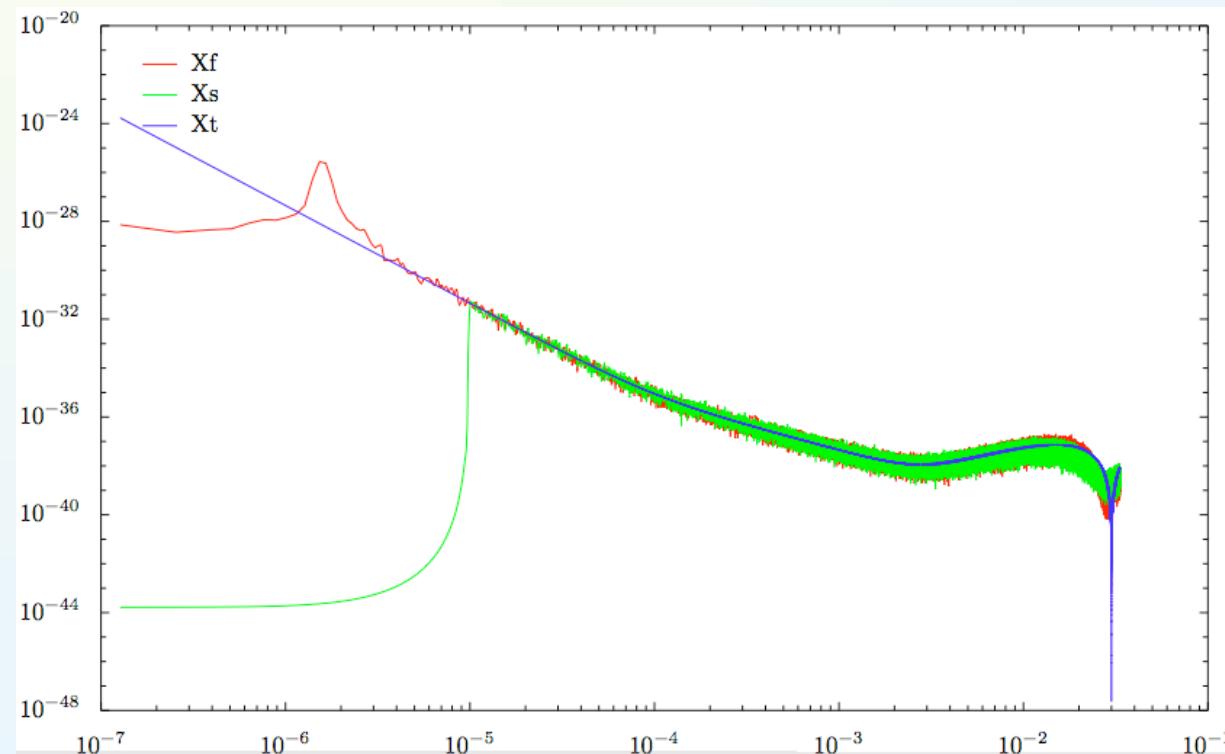
- Keplerian orbits accurate to second order in the eccentricity
- Initial position and rotation of constellation determined by standard param.
- After running through simulators, TDI signals include amplitude and Doppler modulations due to LISA orbits

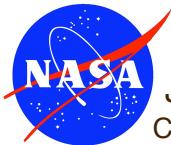




## Noise model

- No laser phase noise in Challenge 1 (assume TDI cancellation)
- White phase optical noise, one-sided  $\text{sqrt}(S_n) = 20 \times 10^{-12} \text{ m Hz}^{-1/2}$
- White+red acceleration noise, one-sided  
 $\text{sqrt}(S_n) = 3 \times 10^{-15} \sqrt{1 + (10^{-4} \text{Hz}/f)^{-2}} \text{ m s}^{-2} \text{ Hz}^{-1/2}$





## TDI observables

- TDI 1.5 (first generation with causal uplink/downlink delays) X, Y, Z
- Expressions from D. A. Shaddock, M. Tinto, F. B. Estabrook, and J. W. Armstrong, Phys. Rev. D **68**, 061303(R) (2003); M. Tinto, F. B. Estabrook, and J. W. Armstrong, Phys. Rev. D **69**, 082001 (2004).
- Coincide with N. J. Cornish and R. W. Hellings, Class. Quantum Grav. **20** 4851 (2003).

$$\begin{aligned} X_1 = & [(s_{31} + s_{13;2}) + (s_{21} + s_{12;3'})_{;2'2} + (s_{21} + s_{12;3'})_{;33'2'2} \\ & + (s_{31} + s_{13;2})_{;33'33'2'2}] - [(s_{21} + s_{12;3'}) \\ & + (s_{31} + s_{13;2})_{;33'} + (s_{31} + s_{13;2})_{2'233'} + (s_{21} + s_{12;3'})_{;2'22'233'}] + \frac{1}{2} [(\tau_{21} - \tau_{31}) - (\tau_{21} - \tau_{31})_{;33'} \\ & - (\tau_{21} - \tau_{31})_{;2'2} + (\tau_{21} - \tau_{31})_{;33'33'2'2} \\ & + (\tau_{21} - \tau_{31})_{;2'22'233'} - (\tau_{21} - \tau_{31})_{;2'233'33'2'2}] \quad (6) \end{aligned}$$

(but check papers)



## XML

- eXtensible Markup Language, a cousin of HTML
- WWW/database/Office industry standard, many RW libraries
- ASCII, based on nested *elements* (w/*attributes*) and *content*
- supports *validation* and *style-sheet transformations*

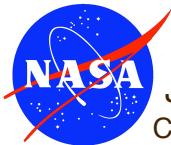


## XML

- eXtensible Markup Language, a cousin of HTML
- WWW/database/Office industry standard, many RW libraries
- ASCII, based on nested *elements* (w/*attributes*) and *content*
- supports *validation* and *style-sheet transformations*

<taskforce>

</taskforce>

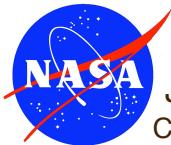


## XML

- eXtensible Markup Language, a cousin of HTML
- WWW/database/Office industry standard, many RW libraries
- ASCII, based on nested *elements* (w/*attributes*) and *content*
- supports *validation* and *style-sheet transformations*

```
<taskforce>
    <member italian="yes">
        Alberto Vecchio
    </member>
```

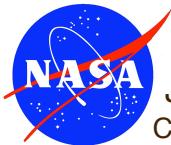
```
</taskforce>
```



## XML

- eXtensible Markup Language, a cousin of HTML
- WWW/database/Office industry standard, many RW libraries
- ASCII, based on nested *elements* (w/*attributes*) and *content*
- supports *validation* and *style-sheet transformations*

```
<taskforce>
    <member italian="yes">
        Alberto Vecchio
    </member>
    <member italian="no">
        Neil Cornish
    </member>
</taskforce>
```



## XML

- eXtensible Markup Language, a cousin of HTML
- WWW/database/Office industry standard, many RW libraries
- ASCII, based on nested *elements* (w/*attributes*) and *content*
- supports *validation* and *style-sheet transformations*

```
<taskforce>
    <member italian="yes">
        Alberto Vecchio
    </member>
    <member italian="no">
        Neil Cornish
    </member>
    <member italian="yes" charming="very">
        Michele Vallisneri
    </member>
</taskforce>
```



## lisaXML

- Based on XSIL: eXtensible Scientific Interchange Language
- Used by LIGO (lightweight data format)

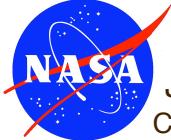


## lisaXML

- Based on XSIL: eXtensible Scientific Interchange Language
- Used by LIGO (lightweight data format)

```
<Param Name="Frequency" Unit="Hertz">  
    1e-3  
</Param>
```

define parameters with  
standard name, multiple units



## lisaXML

- Based on XSIL: eXtensible Scientific Interchange Language
- Used by LIGO (lightweight data format)

```
<Param Name="Frequency" Unit="Hertz">
```

```
    1e-3
```

```
</Param>
```

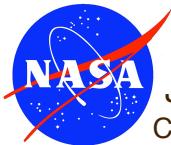
define parameters with  
standard name, multiple units

```
<XSIL Type="LISASource" Name="AMCVn">
```

```
    . . .
```

```
</XSIL>
```

hierarchical file structure  
with XSIL *containers*



## lisaXML

- Based on XSIL: eXtensible Scientific Interchange Language
- Used by LIGO (lightweight data format)

```
<Param Name="Frequency" Unit="Hertz">
```

```
    1e-3
```

```
</Param>
```

define parameters with  
standard name, multiple units

```
<XSIL Type="LISASource" Name="AMCVn">
```

```
    . . .
```

```
</XSIL>
```

hierarchical file structure  
with XSIL *containers*

```
<Array Name="X, Y, Z" Type="double">
```

```
    <Dim Name="Length">2097152</Dim>
```

```
    <Dim Name="Records">3</Dim>
```

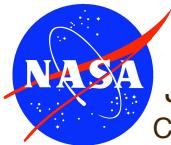
```
    <Stream Type="Remote" Encoding="Binary, BigEndian">
```

```
        amcv.bin
```

```
    </Stream>
```

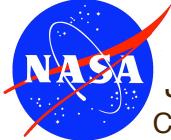
homogeneous array is stored  
efficiently in companion binary

```
</Array>
```



## lisaXML file structure

- Prolog (author, date, comments)
- LISA geometry description
  - Initial orbital and rotation phases for pseudo-LISA model
- LISA noise description
  - Pseudo-random processes with white phase (optical), white + red acceleration (proof mass) PSDs
- GW source description
  - Galactic binary, SMBH binary, EMRI; standard parametrization for model waveforms
  - Sampled  $h_+$ ,  $h_x$
- TDI output
  - Phase-equivalent strain (LISA Simulator), fractional frequency fluctuations (Synthetic LISA)



## XSL magic with a browser

```
<XSIL Name="test-binary-1" Type="PlaneWave">
  <Param Name="EclipticLatitude" Unit="Radian">0.0</Param>
  <Param Name="EclipticLongitude" Unit="Radian">0.0</Param>
  <Param Name="Polarization" Unit="Radian">0.0</Param>

  <Param Name="SourceType">GalacticBinary</Param>

  <Param Name="Amplitude" Unit="1">1.0e-24</Param>
  <Param Name="Inclination" Unit="Radian">0.0</Param>
  <Param Name="TimeOffset" Unit="Second">0.0</Param>
  <Param Name="InitialPhase" Unit="Radian">0.0</Param>
  <Param Name="Frequency" Unit="Hertz">1.0e-3</Param>
</XSIL>
```



Jet Pro  
California

## Mock LISA Data Challenge XML File Format, v

http://www.vallis.org/xml-format.xml

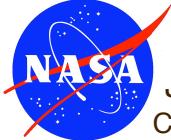
Mock LISA Data Challenge ...

```
<XSIL Name="LISA-1" Version="1.0">
  <Param Name="SourceID" Value="test-binary-1" Type="String"/>
  <Param Name="SourceType" Value="GalacticBinary" Type="String"/>
  <Param Name="EclipticLatitude" Value="0.0" Type="Double"/>
  <Param Name="EclipticLongitude" Value="0.0" Type="Double"/>
  <Param Name="Polarization" Value="0.0" Type="Double"/>
  <Param Name="Amplitude" Value="1.0e-24" Type="Double"/>
  <Param Name="Inclination" Value="0.0" Type="Double"/>
  <Param Name="TimeOffset" Value="0.0" Type="Double"/>
  <Param Name="InitialPhase" Value="0.0" Type="Double"/>
  <Param Name="Frequency" Value="1.0e-3" Type="Double"/>
  <Param Name="FrequencyDot" Value="0.0" Type="Double"/>
  <Param Name="FrequencyDotDot" Value="0.0" Type="Double"/>
  <Param Name="Eccentricity" Value="0.0" Type="Double"/>
  <Param Name="LISA-1" Value="1" Type="Double"/>
</XSIL>
```

Source data

test-binary-1 (PlaneWave)		
EclipticLatitude	0.0	Radian
EclipticLongitude	0.0	Radian
Polarization	0.0	Radian
SourceType	GalacticBinary	
Amplitude	1.0e-24	1
Inclination	0.0	Radian
TimeOffset	0.0	Second
InitialPhase	0.0	Radian
Frequency	1.0e-3	Hertz
FrequencyDot	0.0	Hertz/Second
FrequencyDotDot	0.0	Hertz/Second^2
Eccentricity	0.0	1

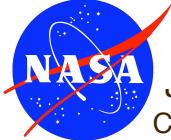
test-BBH-1 (PlaneWave)		
EclipticLatitude	0.0	Radian
EclipticLongitude	0.0	Radian
Polarization	0.0	Radian



Jet Propulsion Laboratory  
California Institute of Technology

- define challenge

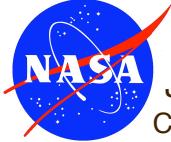
## MLDC workflow by sections in lisaXML files



- define challenge
- generate random parameters

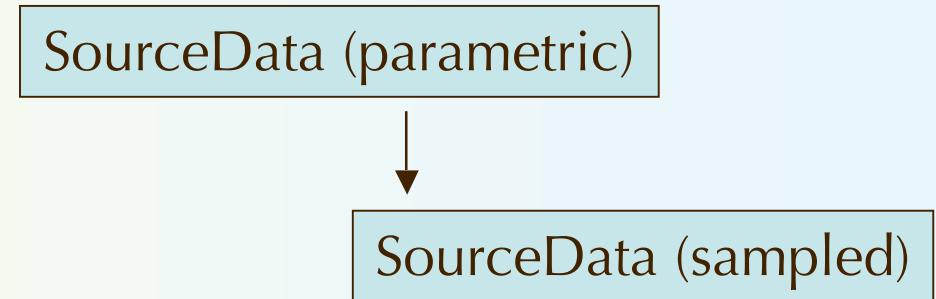
## MLDC workflow by sections in lisaXML files

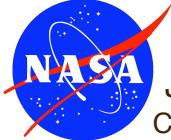
SourceData (parametric)



- define challenge
- generate random parameters
- generate simulator inputs:  
GW strains at SSB

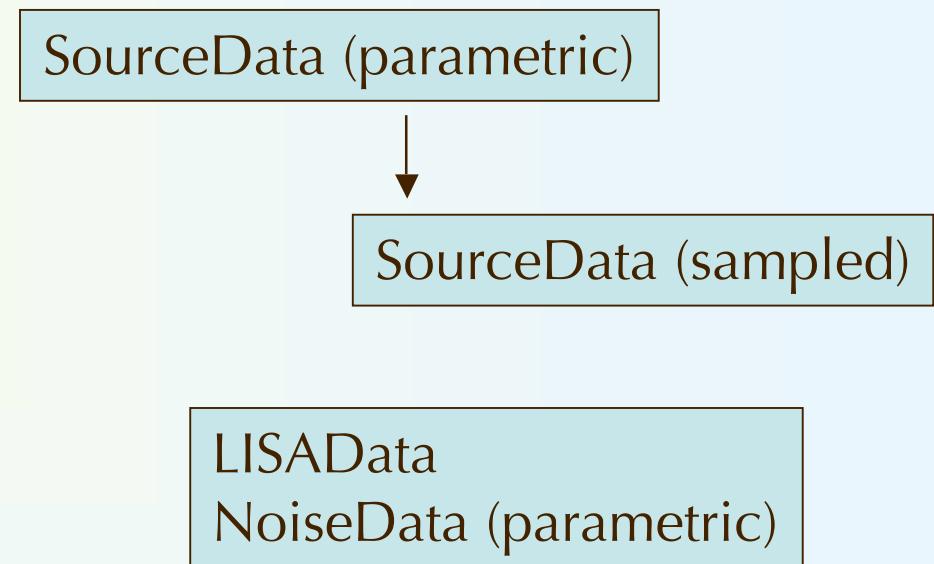
## MLDC workflow by sections in lisaXML files

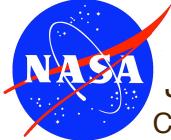




- define challenge
- generate random parameters
- generate simulator inputs:  
GW strains at SSB
- generate pseudorandom  
noise seeds

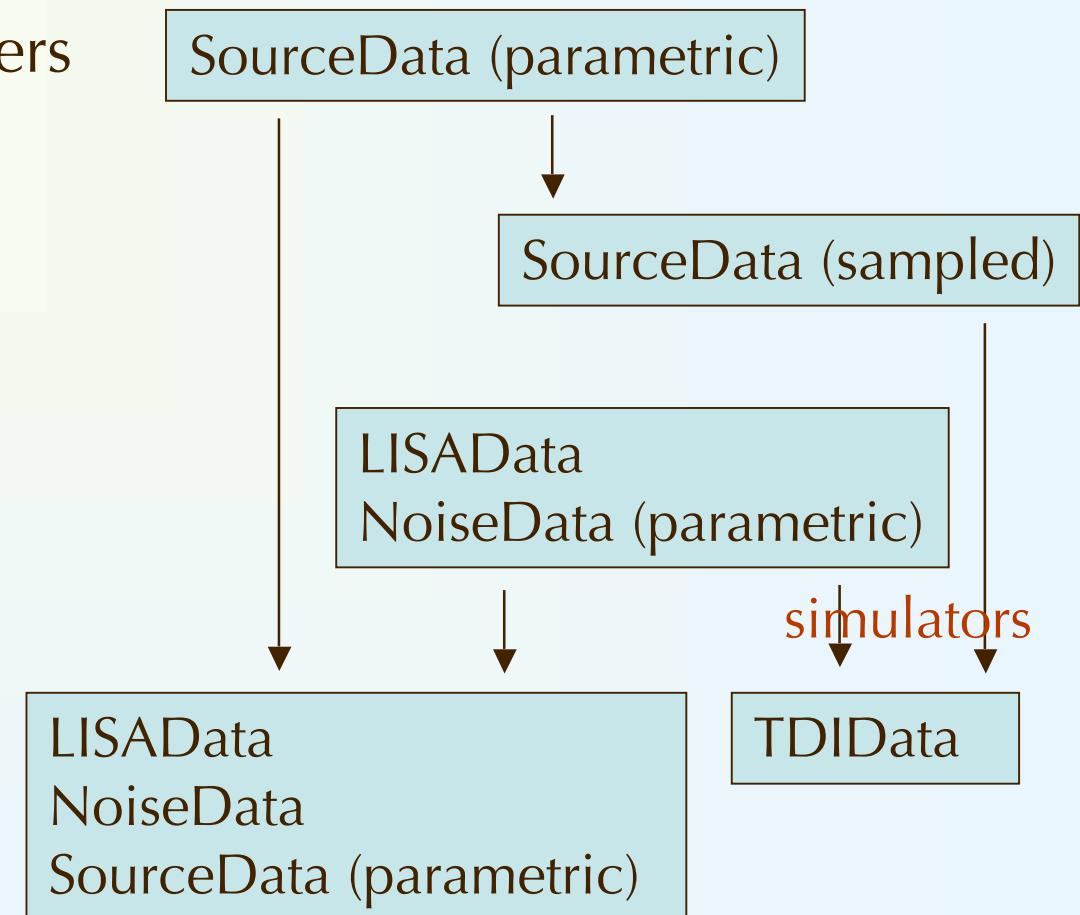
## MLDC workflow by sections in lisaXML files

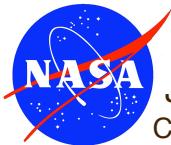




- define challenge
- generate random parameters
- generate simulator inputs:  
GW strains at SSB
- generate pseudorandom  
noise seeds
- generate TDI responses  
(X,Y,Z); save source and  
noise keys

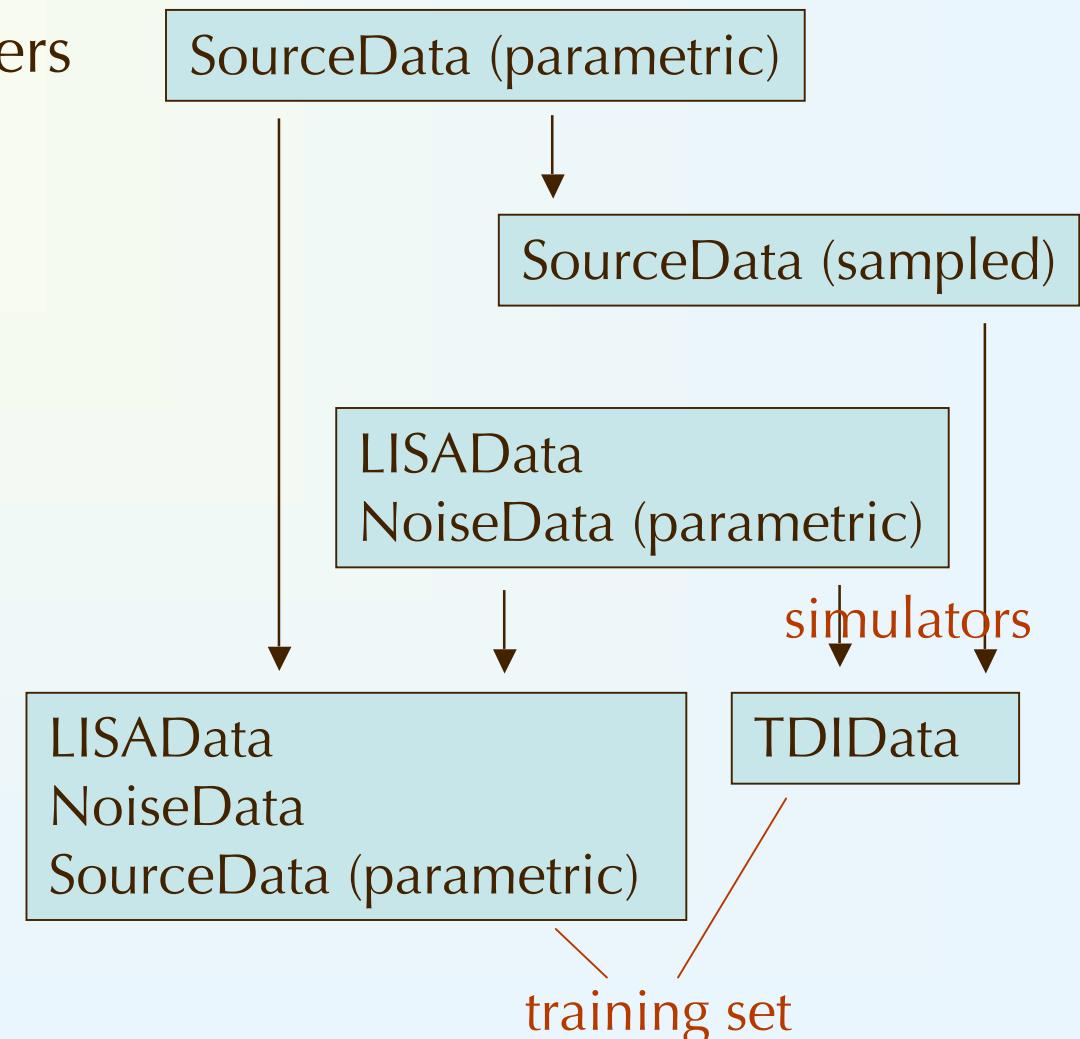
## MLDC workflow by sections in lisaXML files

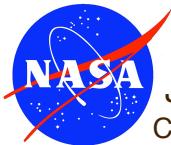




- define challenge
- generate random parameters
- generate simulator inputs:  
GW strains at SSB
- generate pseudorandom  
noise seeds
- generate TDI responses  
(X,Y,Z); save source and  
noise keys

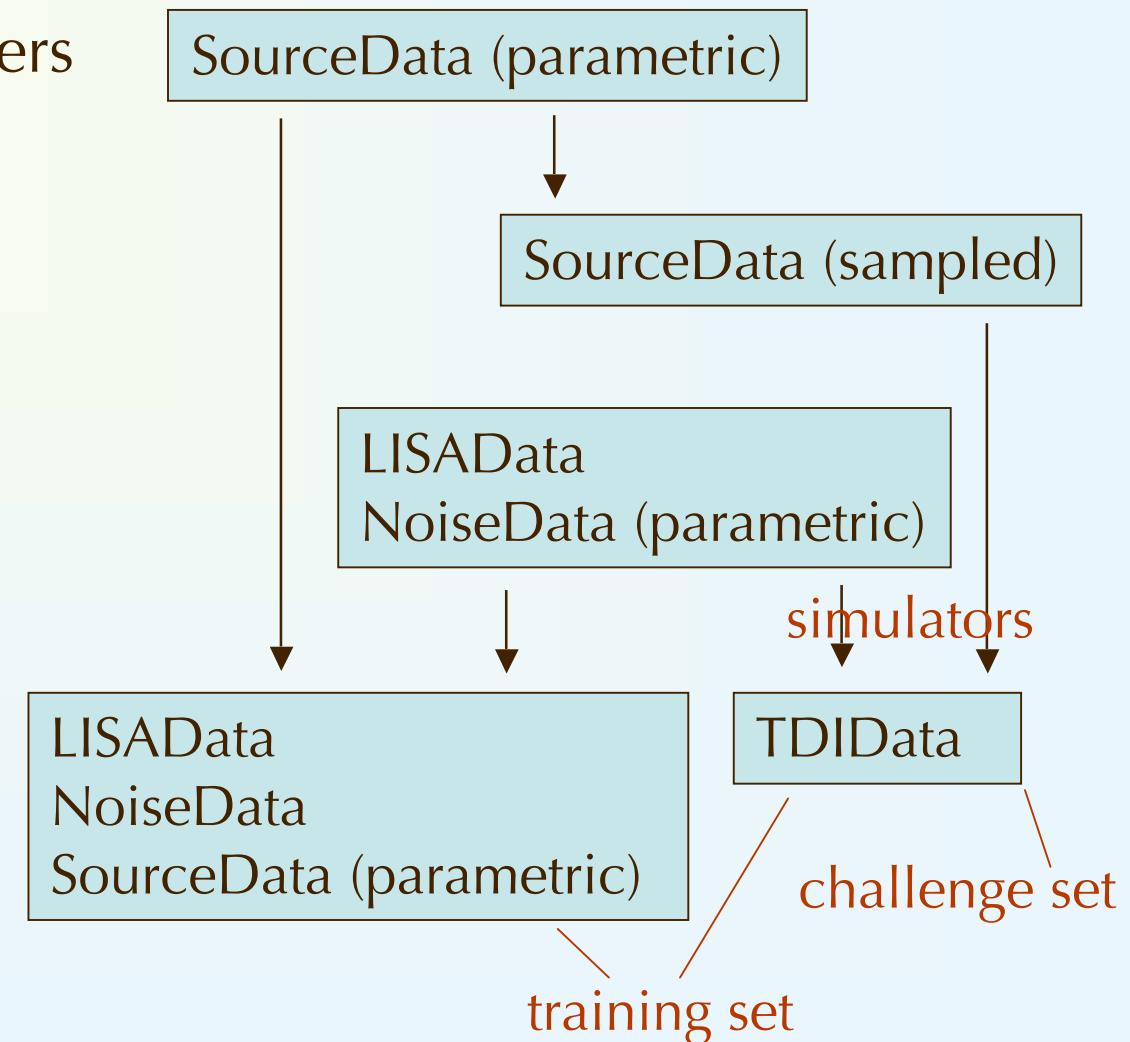
## MLDC workflow by sections in lisaXML files

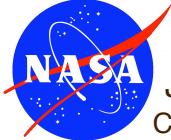




- define challenge
- generate random parameters
- generate simulator inputs:  
GW strains at SSB
- generate pseudorandom  
noise seeds
- generate TDI responses  
(X,Y,Z); save source and  
noise keys

## MLDC workflow by sections in lisaXML files

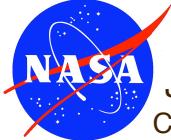




## lisaXML libraries

- C library: lisatools/lisaXML/io-C

```
TimeSeries *ts;  
ts = getTDIdata("challenge.xml");  
printf("' %s ': %d doubles (every %g seconds); ",  
      ts->Name, ts->Length, ts->Cadence);  
/* TDI data is in timeseries->Data[i]->data[] */
```



## lisaXML libraries

- C library: lisatools/lisaXML/io-C

```
TimeSeries *ts;  
ts = getTDIdata("challenge.xml");  
printf("' %s ': %d doubles (every %g seconds); ",  
      ts->Name, ts->Length, ts->Cadence);  
/* TDI data is in timeseries->Data[i]->data[] */
```

- Python library: in Synthetic LISA

```
inputXML = readXML("challenge.xml")  
obs = inputXML.getTDITimeSeries()  
print obs[0]['Cadence']  
X = obs[0]['Data'][:,1]
```



## lisaXML libraries

- C library: lisatools/lisaXML/io-C

```
TimeSeries *ts;  
ts = getTDIdata("challenge.xml");  
printf("' %s ': %d doubles (every %g seconds); ",  
      ts->Name, ts->Length, ts->Cadence);  
/* TDI data is in timeseries->Data[i]->data[] */
```

- Python library: in Synthetic LISA

```
inputXML = readXML("challenge.xml")  
obs = inputXML.getTDITimeSeries()  
print obs[0]['Cadence']  
X = obs[0]['Data'][:,1]
```

- MATLAB: library forthcoming, but: look up binary filename and...

```
readfile = fopen(filename, 'r', 'n');  
obs = fread(readfile, [4, inf], 'double');
```

## Web resources

- MLDC official site: [astrogravs.nasa.gov/docs/mldc](http://astrogravs.nasa.gov/docs/mldc)
- MLDC taskforce wiki:  
[www.tapir.caltech.edu/dokuwiki/listwg1b:home](http://www.tapir.caltech.edu/dokuwiki/listwg1b:home)
- Challenge announce/support mailing list:  
[lisatools-challenge@lists.sourceforge.net](mailto:lisatools-challenge@lists.sourceforge.net)
  - ^ Subscribe at  
<https://lists.sourceforge.net/lists/listinfo/lisatools-challenge>
- MLDC technical support: [sourceforge.net/projects/lisatools/](http://sourceforge.net/projects/lisatools/)
  - ^ Support forums!
  - ^ Subversion (= CVS++) repository for XML libraries, code examples, pipelines, challenge docs
  - ^ Being progressively populated



Jet Propulsion Laboratory  
California Institute of Technology

Let's play!

